Pham, H.H., Khoudour, L., Crouzil, A., Zegers, P. y Velastin, S.A. (2018). Skeletal Movement to Color Map: A Novel Representation for 3D Action Recognition with Inception Residual Networks. In *IEEE International Conference on Image Processing.*

DOI: https://doi.org/10.1109/ICIP.2018.8451404

SKELETAL MOVEMENT TO COLOR MAP: A NOVEL REPRESENTATION FOR 3D ACTION RECOGNITION WITH INCEPTION RESIDUAL NETWORKS

Huy-Hieu Pham[†,††,⋆], *Louahdi Khoudour*[†], *Alain Crouzil*[††], *Pablo Zegers*[†††], *Sergio A. Velastin*[††††,†††††]

[†] Cerema, Equipe-projet STI, 1 Avenue du Colonel Roche, 31400, Toulouse, France
[††] Institut de Recherche en Informatique de Toulouse, Université de Toulouse, UPS, 31062 Toulouse, France
[†††] Aparnix, La Gioconda 4355, 10B, Las Condes, Santiago, Chile
[††††] Applied Artificial Intelligence Research Group, Carlos III University of Madrid, 28270 Madrid, Spain
[†††††] School of Electronic Engineering and Computer Science, Queen Mary University of London, UK

ABSTRACT

We propose a novel skeleton-based representation for 3D action recognition in videos using Deep Convolutional Neural Networks (D-CNNs). Two key issues have been addressed: First, how to construct a robust representation that easily captures the spatial-temporal evolutions of motions from skeleton sequences. Second, how to design D-CNNs capable of learning discriminative features from the new representation in a effective manner. To address these tasks, a skeleton-based representation, namely, SPMF (***S**keleton **P**ose-**M**otion **F**eature*) is proposed. The SPMFs are built from two of the most important properties of a human action: postures and their motions. Therefore, they are able to effectively represent complex actions. For learning and recognition tasks, we design and optimize new D-CNNs based on the idea of Inception Residual networks to predict actions from SPMFs. Our method is evaluated on two challenging datasets including MSR Action3D and NTU-RGB+D. Experimental results indicated that the proposed method surpasses state-of-the-art methods whilst requiring less computation.

***Index Terms*—** Human Action Recognition, SPMF, CNNs.

1. INTRODUCTION

Recognizing correctly actions in untrimmed videos is an important but challenging problem in computer vision. The core difficulties are occlusions, viewpoint, lighting condition and so on [3]. Recently, cost-effective and easy-to-use depth cameras, *e.g.* Microsoft Kinect ™sensor or Intel RealSense, have integrated the real-time skeleton tracking algorithms [4], providing 3D structural information of human motion. This data source is robust to illumination changes, also invariant to camera viewpoints. Thus, exploiting skeletal data for 3D action recognition becomes a very effective research direction. Previous works on Skeleton-based Action Recognition (SAR) can be divided into two main groups: SAR using hand-crafted features and SAR with deep learning networks. The first group [5, 6, 7] uses hand-crafted local features and probabilistic graphical models such as Hidden Markov Model (HMM) [5], Conditional Random Field (CRF) [8], or Fourier Temporal Pyramid (FTP) [7] to classify actions. Almost all of these approaches are data-dependent and require a lot of feature engineering. The second group [9, 10, 11] considers SAR as a time-series problem and proposes the use of Recurrent Neural Networks with Long-Short Term Memory units (RNN-LSTMs) [12] to model the contextual information of the skeletons. Although RNN-LSTMs are able to model the long-term temporal of motion and have advanced the state-of-the-art, this approach just considers skeleton sequences as a kind of the low-level feature, by feeding raw skeletons directly into the network input. The huge number of input features makes RNNs become very complex and may easily lead to overfitting. Moreover, many RNN-LSTMs act as a classifier and can not extract high-level features [13] for recognition tasks.

We believe that an effective representation of motion is the key factor influencing the performance of a SAR model. Different from previous studies, this paper introduces a novel

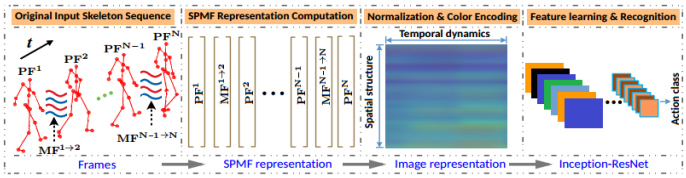

Fig. 1. Schematic overview of our method. Each skeleton sequence is encoded into a color image via a skeleton-based representation called **SPMF**. Each **SPMF** is built from pose vectors (**PF**s) and motion vectors (**MF**s). They are then fed to a D-CNN, which is designed based on the combining of Residual learning [1] and Inception architecture [2] for learning discriminative features from color-coded SPMFs and performing action classification.

⋆This work was done when H. Pham was a visiting scholar at Carlos III University of Madrid. The authors would like to thank the financial support provided by the Cerema Research Center and Institut de Recherche en Informatique de Toulouse, Université de Toulouse, UPS, France for this research.

representation based on skeletal data for 3D action recognition with D-CNNs. To best represent the characteristics of 3D actions, we exploit body poses (***Pose Features*** - **PF**s) and their motions (***Motion Features*** - **MF**s) for building a new representation called **SPMF** (***Skeleton Pose-Motion Feature***). Each SPMF contains important characteristics related to the spatial structure and temporal dynamics of skeletons. A new deep framework based on the Inception Residual networks (Inception-ResNets [14]) is then proposed for learning and classifying actions, as shown in Figure 1. Generally speaking, four hypotheses motivate our exploration of using D-CNNs for SAR are: **(1)** human actions can be correctly represented via the movement of skeletons; **(2)** the spatial-temporal dynamics of skeletons can be transformed into a 2D image structure – a representation that can be effectively learned by learning methods as D-CNNs; **(3)** compared to RGB and depth modalities, skeletons are high-level information while with much less complexity than RGB-D sequences, making action learning model much simpler and faster; (**4**) a well-designed and deeper CNN can improve learning accuracy. Our experimental results on two benchmark datasets confirmed these statements.

The main contributions of this paper can be summarized as follows: **First**, we investigate and propose a novel skeleton-based representation, namely SPMF, for 3D action recognition. **Second**, we design and optimize new D-CNNs based on Inception-ResNet for learning motion features from SPMF in an end-to-end learning framework. To the best of our knowledge, this is the first time a very deep network as Inception-ResNet is exploited for SAR. **Last**, the proposed method set a new state-of-the-art on the MSR Action3D and the NTU-RGB+D datasets. In the next section, we introduce the SPMF representation and the proposed deep learning networks. Experiments and their results are provided in Section 3. Section 4 concludes the paper and discusses the future work.

2. PROPOSED METHOD

2.1. SPMF: From skeleton movement to color map

Two key elements to determine an action are static postures and their motions. We propose SPMF, a novel representation based on these features that are extracted from skeletons. Note that, combining too many geometric features will lead to lower performance than using only a single feature or several main features [15]. In our study, each SPMF is built from pose and motion vectors, as described below:

2.1.1. Pose Feature (PF)

Given a skeleton sequence $\mathcal{S}$ with N frames, denoting as $\mathcal{S} = \{\mathbf{F}^t\}$ with $t \in [1, N]$. Let $\mathbf{p}_j^t$ and $\mathbf{p}_k^t$ be the 3D coordinates of the j-th and k-th joints in each skeleton frame $\mathbf{F}^t$ ($j \neq k$). The **J**oint-**J**oint **E**uclidean distance $\mathbf{JJD}_{jk}^t$ between $\mathbf{p}_j^t$ and $\mathbf{p}_k^t$ at timestamp t is given by:

$$\mathbf{JJD}_{jk}^t = ||\mathbf{p}_j^t - \mathbf{p}_k^t||_2, j \neq k, t \in [1, N] \quad (1)$$

The joint distances obtained by Eq. (1) for the whole dataset range from $\mathbf{D}_{\min} = \mathbf{0}$ to $\mathbf{D}_{\max} = \max\{\mathbf{JJD}_{jk}^t\}$ and noted by $\mathcal{D}_{\textbf{original}}$. The 3D action features can be learned directly from $\mathcal{D}_{\textbf{original}}$ by D-CNNs. However, as $\mathcal{D}_{\textbf{original}}$ is high-dimensional parametric input space, it may lead D-CNNs to time-consuming and overfitting. Therefore, we normalize $\mathcal{D}_{\textbf{original}}$ to the range $[\mathbf{0}, \mathbf{1}]$, denote as $\mathcal{D}_{[\mathbf{0},\mathbf{1}]}$ and considering each distance value in $\mathcal{D}_{[\mathbf{0},\mathbf{1}]}$ as a pixel of the intensity information in a greyscale image. To reflect the change in joint distances, we encode $\mathcal{D}_{[\mathbf{0},\mathbf{1}]}$ into a color space by a sequential discrete color palette. The encoding process is performed by 256-color JET[1] scale. The use of a sequential discrete color palette helps us to reduce the dimensionality of input space that accelerates the convergence rate of deep networks during the representation learning later.

The **J**oint-**J**oint **O**rientation $\mathbf{JJO}_{jk}^t$ from joint $\mathbf{p}_j^t$ to $\mathbf{p}_k^t$ at timestamp t, represented by the unit vector $\overrightarrow{\mathbf{p}_j^t\mathbf{p}_k^t}$ as follows:

$$\mathbf{JJO}_{jk}^t = \text{unit}(\mathbf{p}_j^t - \mathbf{p}_k^t) = \frac{\mathbf{p}_j^t - \mathbf{p}_k^t}{\mathbf{JJD}_{jk}^t}, j \neq k, t \in [1, N] \quad (2)$$

Each vector $\mathbf{JJO}_{jk}^t$ is a 3D vector where all of its components can be normalized to the range $[\mathbf{0}, \mathbf{255}]$. We consider three components (x, y, z) of $\mathbf{JJO}_{jk}^t$ after normalization as the corresponding three components (R, G, B) of a color pixel and define a human pose at timestamp t by vector $\mathbf{PF}^t$ that describes the distance and orientation relationship between joints as a color matrix[2]. Each $\mathbf{PF}^t$ is obtained by:

$$\mathbf{PF}^t = \left[\mathbf{JJD}_{jk}^t\ \mathbf{JJO}_{jk}^t\right], j \neq k, t \in [1, N] \quad (3)$$

2.1.2. Motion Feature (MF)

Let $\mathbf{p}_j^t$ and $\mathbf{p}_k^{t+1}$ be the 3D coordinates of the j-th and k-th joints at two frames consecutively t and $t+1$. Similarly to $\mathbf{JJD}_{jk}^t$ in Eq. (1), the **J**oint-**J**oint **E**uclidean distance $\mathbf{JJD}_{jk}^{t,t+1}$ between $\mathbf{p}_j^t$ and $\mathbf{p}_k^{t+1}$ is computed as:

$$\mathbf{JJD}_{jk}^{t,t+1} = ||\mathbf{p}_j^t - \mathbf{p}_k^{t+1}||_2, t \in [1, N-1] \quad (4)$$

Also, the **J**oint-**J**oint **O**rientation $\mathbf{JJO}_{jk}^{t,t+1}$ from joint $\mathbf{p}_j^t$ to $\mathbf{p}_k^{t+1}$, represented by the unit vector $\overrightarrow{\mathbf{p}_j^t\mathbf{p}_k^{t+1}}$ as follows:

$$\mathbf{JJO}_{jk}^{t,t+1} = \text{unit}(\mathbf{p}_j^t - \mathbf{p}_k^{t+1}) = \frac{\mathbf{p}_j^t - \mathbf{p}_k^{t+1}}{\mathbf{JJD}_{jk}^{t,t+1}}, t \in [1, N-1] \quad (5)$$

[1] A JET color map is based on the order of colors in the spectrum of visible light, ranging from blue to red, passes through the cyan, yellow, orange.

[2] To concatenate vectors of unequal lengths, we expand shorter vectors with neighborhood values until reaching the length of the larger vectors. Before that, all duplicated distances or zero vectors are removed.

We represent a motion from frame t to $t+1$ by vector $\mathbf{MF}^{t\rightarrow t+1}$, given by:

$$\mathbf{MF}^{t\rightarrow t+1} = \left[\mathbf{JJD}_{jk}^{t,t+1}\ \mathbf{JJO}_{jk}^{t,t+1}\right], t \in [1, N-1] \quad (6)$$

where $\mathbf{JJD}_{jk}^{t,t+1}$ and $\mathbf{JJO}_{jk}^{t,t+1}$ are encoded to color matrices as $\mathbf{JJD}_{jk}^{t}$ and $\mathbf{JJO}_{jk}^{t}$. This process not only enhances the texture information in image representation, but also distinguishes the slower and faster motions based on their colors.

2.1.3. Modeling human action with PFs and MFs

Based on the PFs and MFs, we propose a novel skeleton-based representation, namely **SPMF** (***S**keleton **P**ose-**M**otion **F**eatures*). To build the SPMFs, the PFs and MFs are concatenated into a single feature vector by temporal order. Specifically, a SPMF represents a skeleton sequence $\mathcal{S}$, without dependence on the range of actions, is defined as:

$$\begin{aligned}\mathbf{SPMF}_{\mathcal{S}} = [&\mathbf{PF}^1\ \mathbf{MF}^{1\rightarrow 2}\ \mathbf{PF}^2\ ...\ \mathbf{PF}^t\ \mathbf{MF}^{t\rightarrow t+1}\ \mathbf{PF}^{t+1}\\ &...\ \mathbf{PF}^{N-1}\ \mathbf{MF}^{N-1\rightarrow N}\ \mathbf{PF}^N], t \in [2, N-2]\end{aligned} \quad (7)$$

Figure 2 shows some SPMFs obtained from the MSR Ac-tion3D dataset [16] after resizing them to 32×32 pixels.

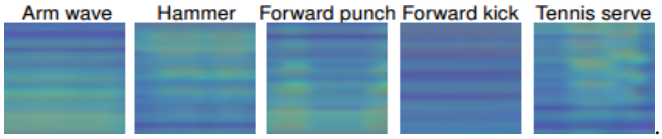

Fig. 2. The SPMFs obtained from some samples of the MSR Action3D dataset. Color-changing reflects the change in distance between skeleton joints. Best viewed in color.

2.2. Inception Residual learning

D-CNNs have demonstrated state-of-the-art performance on many visual recognition tasks. In particular, the recent Inception architecture [2] significantly improved both the accuracy and computational cost through three key ideas: (**1**) reducing the input dimension; (**2**) increasing not only the network depth, but also its width and (**3**) concatenating feature maps learned by different layers. However, very deep networks as Inception are very difficult to train due to the vanishing problem and degradation phenomenon [17]. To this end, ResNet [1] has been introduced. The key idea is to improve the flow of information and gradients through layers by using identity connections. A layer, or a sequence of layers of a traditional CNN learns to calculate a mapping function $y = \mathcal{F}(x)$ from the input feature x. Meanwhile, a ResNet building block approximately calculates the function $y = \mathcal{F}(x) + id(x)$ where $id(x) = x$. This idea helps the learning process to be faster and more accurate. To learn spatio-temporal features from the SPMFs, we propose the combination of Residual learning [1] and Inception architecture [2] to build D-CNNs[3]. Batch normalization [18] and Exponential Linear Units (ELUs) [19] are applied after each Convolution. Dropout [20] with a rate of **0.5** is used to prevent overfitting. A Softmax layer is employed for classification task. Our networks can be trained in an end-to-end manner by the gradient decent using Adam update rule [21]. During training, our goal is to minimize the cross-entropy loss function between the ground-truth label $\mathbf{y}$ and the predicted label $\hat{\mathbf{y}}$ by the network over the training sam-ples $\mathcal{X}$, which is expressed as follows:

$$\mathcal{L}_{\mathcal{X}}(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{M}\left(\sum_{i=1}^{M}\left(\sum_{j=1}^{C}\mathbf{y}_{ij}\log\hat{\mathbf{y}}_{ij}\right)\right) \quad (8)$$

where M indicates the number of samples in training set $\mathcal{X}$ and C denotes the number of action classes.

3. EXPERIMENTS

3.1. Datasets and settings

The proposed method is evaluated on the MSR Action3D and NTU-RGB+D datasets[4]. We follow the evaluation protocols as provided in the original papers. The performance is measured by average classification accuracy over all action classes.

MSR Action3D dataset [16]: This Kinect 1 captured dataset contains 20 actions, performed by 10 subjects. Each skeleton has **20** joints. Our experiments were conducted on 557 action sequences in which the whole dataset is divided into three subsets: Action Set 1 (**AS1**), Action Set 2 (**AS2**), and Action Set 3 (**AS3**). For each subset, a half of the data is selected for training and the rest for testing. More details are provided in the Supplementary Material.

NTU-RGB+D dataset [11]: This Kinect 2 captured dataset is currently the largest dataset for SAR, also very challenging due to its large intra-class and multiple viewpoints. The NTU-RGB+D provides more than 56,000 videos, collected from 40 subjects for 60 action classes. Each skeleton contains the 3D coordinates of **25** body joints. Two different evaluation criteria have been suggested. For the **Cross-Subject** evaluation, the sequences performed by 20 subjects are used for training and the rest sequences are used for testing. For **Cross-View** evaluation, the sequences provided by cameras **2** and **3** are used for training while sequences from camera **1** are used for testing.

3.2. Implementation details

Three different network configurations were implemented and evaluated in Python with Keras framework using the Tensor-

[3] Please refer to the Supplementary Materials to see details of the proposed network architectures.

[4] The list of action classes is provided in Supplementary Materials.

Table 1. Accuracy rate (%) on the MSR Action3D dataset.

Method (protocol of [16])	AS1	AS2	AS3	Aver.
Bag of 3D Points [16] Depth	72.90	71.90	71.90	74.70
Motion Maps [23] Lie Group	96.20	83.20	92.00	90.47
Representation [7]	95.29	83.87	98.22	92.46
Hierarchical RNN [10]	**99.33**†	94.64	95.50	94.49
ST-LSTM Trust Gates [9]	N/A	N/A	N/A	94.80
Graph-Based Motion [24] ST-	93.60	95.50	95.10	94.80
NBNN [25]	91.50	95.60	97.30	94.80
S-T Pyramid [26] Ensemble	99.10	92.90	96.40	96.10
TS-LSTM v2 [27]	95.24	96.43	**100.0**	97.22
SPMF Inception-ResNet-121‡	97.06	**99.00**	98.09	**98.05**
SPMF Inception-ResNet-222	97.54	**98.73**	99.41	**98.56**
SPMF Inception-ResNet-242	96.73	**97.35**	98.77	**97.62**

†
Results that outperform previous works are in **bold**, best accuracies are in **bold-blue**.
‡Denote the number of building blocks Inception-ResNet-A, Inception-ResNet-B, and Inception-ResNet-C, respectively. Details are provided in Supplementary Materials.

Flow backend. During training, we use mini-batches of **256** images for all networks. The weights are initialized by the He initialization technique [22]. Adam optimizer [21] is used with default parameters, $\beta_1 = \mathbf{0.9}$ and $\beta_2 = \mathbf{0.999}$. The ini-tial learning rate is set to **0.001** and is decreased by a factor of **0.5** after every **20** epochs. All networks are trained for **250** epochs from scratch. We applied some simple data aug-mentation techniques (*i.e.*, randomly cropping, flipping and Gaussian filtering) on the MSR Action3D dataset [16] due to its small size. For the NTU-RGB+D [11], we do not apply any data augmentation method.

3.3. Results and comparison with the state-of-the-art

Table 1 reports the experimental results and comparisons with state-of-the-art methods on the MSR Action3D dataset [16]. We achieved the best recognition accuracy by SPMF Inception-ResNet-222 network configuration with a total average accu-racy of **98.56**%. This result outperforms many previous stud-ies [16, 23, 7, 10, 9, 24, 25, 26, 27]. For the NTU-RGB+D dataset [11], we achieved an accuracy of **78.89**% on cross-subject evaluation and **86.15**% on cross-view evaluation as shown in Table 2. These results are better than previous state-of-the-art works reported in [7, 10, 9, 11, 28, 29].

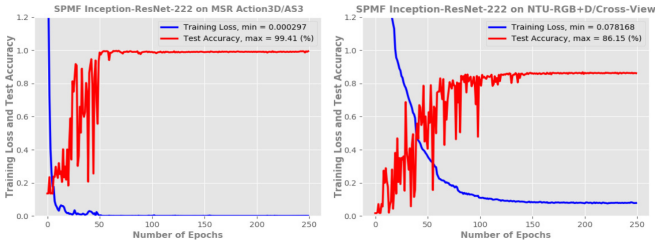

Fig. 3. Training loss and test accuracy of SPMF-Inception-ResNet-222 on MSR Action3D and NTU-RGB+D datasets.

Table 2. Accuracy rate (%) on the NTU-RGB+D dataset.

Method (protocol of [11])	Cross-Subject	Cross-View
Lie Group Representation [7]	50.10	52.80
Hierarchical RNN [10]	59.07	63.97
ST-LSTM Trust Gates [9] Two-	69.20	77.70
Layer P-LSTM [11] Dynamic	62.93	70.27
Skeletons [28] STA-LSTM [30]	60.20	65.20
Depth and Skeleton Fusion [29]	73.40	81.20
	75.20	83.10
SPMF Inception-ResNet-121	**77.02**	82.13
SPMF Inception-ResNet-222	**78.89**	**86.15**
SPMF Inception-ResNet-242	**77.24**	**83.45**

Fig. 4. Visualizing intermediate features generated by Inception-ResNet-222 after feeding several SPMFs into the network. These SPMFs come from samples in the MSR Action3D dataset [16]. Best viewed in color.

3.4. Training and prediction time

We take the NTU-RGB+D dataset with cross-view settings and SPMF-Inception-ResNet-222 network for illustrating the computational efficiency of our learning framework. With the implementation in Python on a single GeForce GTX Ti GPU, no parallel processing, the training phase takes $\mathbf{1.85 \times 10^{-3}}$ second per sequence. While the testing phase, including the time for encoding skeletons into color images and classification, takes **0.128** second per sequence. This speed is fast enough to the needs of many real-time applications. These results verify the effectiveness of the proposed method, not only in terms of accuracy, but also in terms of computational cost.

4. CONCLUSION AND FUTURE WORK

In this paper, we introduced a new method for recognizing human actions from skeletal data. A novel skeleton-based representation, namely SPMF, is proposed for encoding spatial-temporal dynamics of skeleton joints into color images. Deep convolutional neural networks based on Inception Residual architecture are then exploited to learn and recognize actions from obtained image-based representations. Experiments on two publicly available benchmark datasets have demonstrated the effectiveness of the proposed representation as well as feature learning networks. We are currently expanding this study by improving the color encoding algorithm through an image enhancement method in order to generate more discriminative features contained in the image representations.

5. REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[2] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *CVPR*, 2016.

[3] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, 2010.

[4] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, 2013.

[5] F. Lv and R. Nevatia, "Recognition and segmentation of 3-D human action using HMM and multi-class AdaBoost," in *ECCV*, 2006.

[6] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *CVPR*, 2012.

[7] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," in *CVPR*, 2014.

[8] L. Han, X. Wu, W. Liang, G. Hou, and Y. Jia, "Discriminative human action recognition in the learned hierarchical manifold space," *Image and Vision Computing*, vol. 28, 2010.

[9] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *ECCV*, 2016.

[10] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *CVPR*, 2015.

[11] A. Shahroudy, J. Liu, T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *CVPR*, 2016.

[12] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, 1997.

[13] T. N. Sainath, O. Vinyals, A. W. Senior, and H. Sak, "Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks," in *ICASSP*, 2015.

[14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the impact of Residual connections on learning," in *AAAI*, 2017.

[15] S. Zhang, X. Liu, and J. Xiao, "On geometric features for skeleton-based action recognition using multilayer LSTM networks," in *WACV*, 2017.

[16] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *CVPR*, 2010.

[17] K. He and J. Sun, "Convolutional Neural Networks at Constrained Time Cost," in *CVPR*, 2015.

[18] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[19] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by Exponential Linear Units (ELUs)," *arXiv preprint arXiv:1511.07289*, 2015.

[20] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[21] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *ICCV*, 2015.

[23] C. Chen, K. Liu, and N. Kehtarnavaz, "Real-time human action recognition based on depth motion maps," *Journal of Real-time Image Processing*, vol. 12, 2016.

[24] P. Wang, C. Yuan, W. Hu, B. Li, and Y. Zhang, "Graph based skeleton motion representation and similarity measurement for action recognition," in *ECCV*, 2016.

[25] J. Weng, C. Weng, and J. Yuan, "Spatio-Temporal Naive-Bayes Nearest-Neighbor (ST-NBNN) for skeleton-based action recognition," in *CVPR*, 2017.

[26] H. Xu, E. Chen, C. Liang, L. Qi, and L. Guan, "Spatio-Temporal Pyramid Model based on depth maps for action recognition," in *MMSP*, 2015.

[27] C. Li, P. Wang, S. Wang, Y. Hou, and W. Li, "Skeleton-based action recognition using LSTM and CNN," in *ICMEW*, 2017.

[28] J. Hu, W. Zheng, J. Lai, and Z. Jianguo, "Jointly learning heterogeneous features for RGB-D activity recognition," in *CVPR*, 2015.

[29] H. Rahmani and M. Bennamoun, "Learning action recognition model from depth and skeleton videos," in *ICCV*, 2017.

[30] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *AAAI*, 2017.